

NICS - Galileo Near Infrared Camera  
Programma di controllo per il DSP sequencer

G.Comoretto, C. Baffa

Osservatorio Astrofisico di Arcetri, l.go E. Fermi 5, 50125 Firenze (Italy)

**Arcetri Technical Report N° 4/1998**  
**Firenze, Aprile 1998**

*Il controller per le camere CCD del progetto Galileo comprende un DSP 56001, che ha il compito di generare tutte le forme d'onda richieste, e di controllare le temporizzazioni.*

*Sebbene il sistema sia stato disegnato con la massima generalità possibile, le caratteristiche specifiche del multiplexer Hawaii utilizzato nella camera NICS richiede delle modifiche al programma di controllo del DSP. In particolare, è necessario modificare lo schema di lettura, che è radicalmente differente da quello di un CCD ottico, e introdurre la possibilità di un readout multiplo durante la misura.*

## 1 Specifiche di progetto

Il controller per CCD sviluppato dal CCD Working Group è descritto in dettaglio ad es. in [2], e il suo uso relativo alla camera NICS è descritto in [4].

Riassumiamo qui brevemente i principi di funzionamento del controller. Questo controller è basato su un modulo transputer contenente un DSP Motorola 56001, che accede a periferiche generiche attraverso un bus dati/indirizzi, e che può pilotare direttamente 16+12 linee digitali. Attraverso una logica dedicata, è possibile generare sequenze arbitrarie di clock su queste linee, che vengono utilizzate per pilotare il chip CCD. La lettura del chip viene fatta attraverso convertitori ADC collegati ad uno o più transputer, che comunicano con il resto del sistema attraverso link transputer standard.

Ulteriori funzioni di controllo e telemetria vengono svolte attraverso periferiche interfacciate al bus del DSP.

Nella gestione del CCD, pertanto, il DSP deve eseguire un numero molto limitato di compiti, e cioè

- Generazione delle forme d'onda per la scansione del CCD e la temporizzazione dell'acquisizione
- Generazione del tempo di integrazione
- Programmazione dei convertitori DAC che generano i livelli di bias nella scheda analogica
- Lettura del convertitore ADC per la telemetria interna della scheda analogica

Oltre a questo, è naturalmente necessario disporre di funzioni generiche di test sia inteno al DSP che del bus del sequencer, che questi controlla.

La funzione fondamentale svolta è quella della temporizzazione delle forme d'onda necessarie per l'acquisizione. Per garantire la massima flessibilità al controller, queste sono programmate utilizzando tabelle, generate con un software specifico nel formato Ogni tabella descrive le forme d'onda generate in una delle due porte decodificate del CCDBUS, che serve ad implementare una funzione elementare nel CCD (es. lettura di un pixel). Il DSP viene programmato con i parametri dell'area del CCD da leggere, e combina tra di loro le forme d'onda in modo opportuno per eseguire una lettura completa.

In un sensore infrarosso, difficilmente si deve eseguire una singola lettura, Occorre prevedere pertanto un insieme di modi di funzionamento, eventualmente estendibile, e selezionabile tramite un parametro. Per ora si individuano due modi di funzionamento:

- Integrazione singola, con e senza reset. Serve essenzialmente per debug del rivelatore.
- Integrazione elementare con doppio campionamento. La prima lettura è fatta subito dopo il reset, dopo ciascuna integrazione viene effettuata una lettura-reset-lettura, e dopo l'ultima una lettura semplice. E' necessario specificare il tempo di integrazione elementare ed il numero di integrazioni da effettuare. Può essere utile specificare un numero di cicli di reset  $> 1$ , con lettura del valore di zero solo dopo l'ultimo di questi.
- Integrazione con campionamento multiplo non distruttivo durante l'integrazione. E' praticabile solo con tempi di integrazione lunghi. Richiede si specifichi il numero di punti campionati durante un'integrazione, insieme al numero di integrazioni elementari.

## 2 Comunicazione con il DSP

Il DSP comunica con un transputer T20 posto nel modulo Perimos DTM560. La comunicazione avviene per mezzo di una serie di routines assembler fornite dalla Perimos, utilizzando un protocollo sviluppato dal CCD Working Group.

Il grosso del software del DSP realizza questo protocollo, ed interpreta i comandi definiti in questo. E' stato tuttavia necessario aggiungere e/o modificare alcuni comandi, per tener conto delle differenze dell'hardware collegato al bus CCDSEQ della camera NICS. In questo capitolo si descrive in dettaglio il protocollo originale, e le modifiche effettuate.

### 2.1 Protocollo di comunicazione standard CCDWG

IL CCDWG ha standardizzato un formato di comandi per il DSP, assieme ad un insieme standard di comandi. Ogni comando genera una *telemetria*, cioè un pacchetto di dati in risposta, che segnala che il comando è stato eseguito, ed eventualmente un insieme di valori prodotti in risposta.

Ogni comando ha la stessa struttura, come pacchetto di 40 word a 24 bit. Dal lato del transputer, le word sono a 32 bit, ma gli 8 bit piu' significativi vengono ignorati. La struttura (espressa in C++) ha la forma:

```
struct cmdbuffer {
    word    opcode; // Codice (vedi tab. 1)
    word    flags; // non usati
    word    destination; // codice di identificazione
            // del DSP nel sistema
    word    opnum; // numero di parametri
    word    operand[36]; }; // parametri
```

Al termine dell'esecuzione, il DSP genera una struttura analoga, con la forma:

```
struct tlmbuffer {
    word tag;
    word flags;
    word sender;
    word opnum;
    word operand[36]; };
```

I componenti della struttura hanno lo stesso significato che nella struttura `cmdbuffer`, e, nel caso in cui il comando non abbia bisogno di ritornare informazioni, è semplicemente una copia identica della struttura stessa. Nel caso in cui invece il comando debba ritornare informazione, vengono variati i campi `opnum` e `operand`, che conterranno rispettivamente il numero di word ritornate e l'informazione richiesta.

## 2.2 Comandi definiti nel protocollo CCDWG

I comandi riconosciuti nella versione standard del software sono elencati in tab.1.

Code	Function	Parameters
1	Get Telemetry Buffer	
10	Set CCDBUS port A to value	Value
20	Set CCDBUS port B to value	Value
30	Set location on CCDBUS to value	Location, value
40	Test waveform	Iterations, WF ID
50	Loopback	
60	Dump 36 locations from X bus	Start addr.
70	Dump 36 locations from Y bus	Start addr.
110	Return card ID and Sync bit	
120	Toggle debug LED	
200	Set sensor dimensions	X-size, Y-size
210	Set readout box	X and Y-size, X and Y position
220	Load waveform table	ID, port, table (up to 34 values)
230	Start integration	Exposure time, n. of integrations

Tabella 1: Comandi riconosciuti dal software CCDWG del DSP. Nota: tutti i valori sono in base decimale

La maggior parte di questi comandi sono sufficientemente elementari da non richiedere una descrizione dettagliata. Alcune note sono però utili per la maggior parte di essi.

### 2.2.1 Ask telemetry

Questo comando dovrebbe ritornare il valore della *telemetria*. Non è attualmente definito cosa debba essere monitorato con questo comando, che risulta attualmente equivalente ad un loopback. È stato introdotto in quanto nelle specifiche originali del programma era possibile che alcuni comandi venissero eseguiti in modo differito, e che quindi fosse necessario leggerne i risultati in un secondo tempo.

### 2.2.2 Set location

La locazione da modificare è espressa all'interno dello spazio di indirizzamento del bus CCDSEQ (256 locazioni). Nella versione corrente del software, questo spazio è posto sul bus X, a partire dalla locazione 0x200 (512). Pertanto un comando di set location 20,10 imposterà la locazione 532 (512+20) sul bus X al valore 10.

L'inizio dello spazio di indirizzamento è fissato dal valore della costante `MAP_BASE` nel programma assembler, che deve corrispondere all'impostazione degli switch sulla scheda CCDSEQ.

### 2.2.3 Test waveform

Questo comando è utile in fase di debug, per iterare una forma d'onda elementare il numero desiderato di volte. I parametri sono il numero di iterazioni (che deve essere > 0), e il codice identificativo della waveform.

#### 2.2.4 Dump X (Y) bus

Questi comandi leggono il contenuto di 36 locazioni consecutive a partire da una locazione arbitraria sul bus X (o Y) del DSP. È quindi possibile esaminare memoria interna o esterna del DSP, e locazioni sul bus del CCDSEQ. In quest'ultimo caso occorre ricordare che il bus parte dalla locazione 512 (0x200) del bus X, e quindi per esaminare la locazione N del CCDSEQ bus, occorre dare un comando di dump della locazione 512+N. Questo è parzialmente inconsistente con l'uso del comando *set location*.

Occorre inoltre notare che i buffer del CCDSEQ introducono ritardi tali da rendere difficile l'accesso di locazioni sul bus senza introdurre wait states. Pertanto nella versione per NICS sono stati introdotti wait states in software (vedi cap. 2.3.2).

#### 2.2.5 Return card ID and Sync bit

Questo comando ritorna due parametri: il primo è il codice identificativo della scheda, come impostato sullo switch SW2. Il secondo è lo stato del latch di sincronismo. Quest'ultimo è settato da un fronte di salita sul connettore BNC del controller, e viene resettato durante la lettura.

#### 2.2.6 Set sensor dimensions, and readout box

Tutte le dimensioni sono espresse in pixel. La dimensione del sensore indica risp. il numero di righe e di colonne di cui è composto, con la convenzione usuale che la riga è la direzione in cui i pixel vengono estratti in successione.

L'area utile del sensore (readout box) è definita dalle sue dimensioni, e dal numero di righe e di colonne da saltare prima di cominciare la lettura.

#### 2.2.7 Load waveform table

Una waveform è definita da un array di valori da impostare in sequenza su di una porta arbitraria del DSP. Di solito vengono impiegate le porte decodificate A e B (risp. ai primi due indirizzi dello spazio di indirizzamento del CCDSEQ). Ogni valore contiene una parte (16 bit meno significativi) che vengono scritti sulla porta, e una (8 bit più significativi) che pilotano la logica di ritardo del controller.

La logica di ritardo è descritta in dettaglio in una serie di rapporti interni. Qui si richiama brevemente i concetti fondamentali. Il valore di ritardo specificato determina quanti wait states vengono inseriti nel *successivo* accesso allo spazio di indirizzamento del controller, e quindi il tempo *minimo* che la configurazione di bit resta valida sul bus prima di venir sovrascritta. In particolare, determina il tempo esatto che le configurazioni di una waveform, eccetto l'ultima, restano valide.

Un valore di ritardo di zero implica che non vengono inseriti wait states. La parola resta scritta per 100 ns. Un valore N diverso da zero inserisce  $256 - N$  wait states, per un tempo totale di  $(257 - N) * 100$ ns. Un valore di 255 non è valido, e una durata di 200 ns viene ottenuta ripetendo due volte la stessa configurazione senza wait states.

Ogni waveform ha un formato composto da un codice identificativo, l'indirizzo della porta su cui questa viene scritta, nello spazio del buss CCDSEQ, il numero di parole di cui è composta, e la sequenza di valori.

Il software di integrazione identifica le waveform dal codice identificativo. Se due sequenze hanno lo stesso codice, la seconda sovrascrive la prima, ma *non rilascia* la memoria utilizzata dalla prima. La memoria totale disponibile per le sequenze è di 256 word.

Nel software originale, le sequenze, in successione, servono alle seguenti funzioni:

- 10: Scarico della regione sensibile del CCD nel buffer interno

- 20: Avanzamento riga
- 30: Avanzamento colonna senza lettura
- 40: Avanzamento colonna con lettura

Queste operazioni sono specifiche per un CCD ottico, e sono modificate nel software per NICS.

### 2.2.8 Start integration

Questo comando esegue una integrazione. Il comando impegna il DSP, che non può quindi per es. essere interrotto, per tutta la durata dell'integrazione.

Il comando accetta due parametri: il tempo di integrazione elementare, espresso in millisecondi, e il numero di volte che l'integrazione elementare deve essere ripetuta. Il tempo di integrazione massimo è di  $(2^{24} - 1)$  ms (sconsigliato).

Ogni integrazione elementare è composta da un loop di attesa per il tempo specificato, seguito da una sequenza di lettura standard. La sequenza di lettura è specifica per un CCD ottico, e quindi è stata modificata nella versione per NICS.

## 2.3 Aggiunte e modifiche nella versione per NICS

Rispetto a comandi definiti in tabella 1, il software per NICS è stato modificato per gestire le peculiarità dell'hardware utilizzato. L'elenco dei comandi, ed il formato di comandi e telemetria, non è stato comunque modificato.

In particolare, il controller deve impostare alcuni valori di tensione per il bias, eseguire letture di telemetria, e impostare la lunghezza della riga da leggere nei chip di controllo dei convertitori ADC veloci.

Inoltre la sequenza di lettura di un multiplexer infrarosso è completamente diversa rispetto ad un CCD ottico, e non è sufficiente modificare le waveform utilizzate. Sono necessarie un numero maggiore di waveform elementari, e risulta necessario definire alcuni parametri aggiuntivi.

Occorrerebbe inoltre prevedere un comando di abort dell'integrazione, in quanto attualmente il comando di inizio integrazione termina *incondizionatamente* solo al termine dell'integrazione.

La versione attuale del software utilizza il meccanismo di comunicazione tra DSP e T200 nel modulo Perimos previsto dalle funzioni di libreria del modulo stesso. Queste funzioni lavorano in polling mode, cioè il processore non risponde a comandi inviati mentre un comando precedente è in esecuzione, ad es. durante l'integrazione. Risulta quindi complesso, ed al di là dello scopo di questo rapporto, definire in dettaglio un protocollo per gestire la possibilità di abortire un'integrazione.

I comandi che sono stati modificati rispetto alla versione standard del software sono elencati in tab.2:

Code	Function	Parameters
1	Get Telemetry Buffer	
60	Dump 36 locations from X bus	Start addr.
200	Set sensor dimensions	X-size, Y-size, n. output
220	Load waveform table	ID, port, table (up to 34 values)
230	Start integration	Exp. time, n. integr., mode, n.samples

Tabella 2: Comandi modificati, nel significato o nei parametri, nel software CCDWG del DSP

### 2.3.1 Get Telemetry

Il comando inizia una sequenza di lettura dei valori di bias correntemente impostati. Ritorna 8 valori, corrispondenti agli 8 ingressi del convertitore ADC della scheda di condizionamento analogico dei segnali.

### 2.3.2 Dump from X bus

Per permettere la lettura di locazioni poste sul bus del CCDSEQ, la lettura avviene con 2 wait states inseriti, per un totale di 300 ns. In questo modo anche periferiche relativamente lente possono essere lette senza errori. Per introdurre questi wait states si è utilizzata la possibilità del DSP di specificarne un numero arbitrario in un registro interno della CPU.

Non si è introdotto wait states per letture sul bus Y (mappato in memoria).

### 2.3.3 Set sensor dimensions

È stato aggiunto un parametro, che specifica il numero di uscite del sensore. Siccome l'hardware di NICS richiede che il DSP inicializzi la logica di serializzazione dei convertitori ADC veloci, occorre che il DSP sappia *quanti* convertitori siano presenti.

### 2.3.4 Load waveform table

Il comando è invariato, ma sono caricabili 7 waveform indipendenti, con significato differente rispetto al software originale. È inoltre stato corretto un bug che impediva il caricamento di più di 6 waveform. Le waveform definite sono:

- 10: Inizio frame. Inizializza gli shiftregister di riga e colonna
- 20: Avanzamento di una riga
- 30: Inizio riga. Inizializza lo shiftregister di colonna
- 40: Shift veloce di un pixel senza lettura
- 50: Shift di un pixel con lettura
- 60: Inizio riga con reset della stessa
- 70: Non è attualmente utilizzata.

### 2.3.5 Start integration

Sono specificati due ulteriori parametri: un modo osservativo, per permettere di specificare in corsa modi differenti, e un numero di campionamenti per lettura, che serve nei modi a campionamento multiplo durante l'integrazione.

Attualmente questi parametri non sono utilizzati, in quanto è definito un solo modo osservativo, descritto nel cap. 5.5, che non permette un campionamento multiplo durante l'integrazione. Il software comunque è predisposto per funzionare con modi differenti, e questi parametri sono stati previsti per permettere queste modifiche.

## 3 Spazio di indirizzamento del DSP

Il DSP ha 32 KW di memoria esterna e 1 KW di memoria interna, entrambe a 24 bit. Al boot, viene selezionata una mappa di memoria del modulo DTM560 che ripartisce questa memoria come in tab. 3. Il DSP 56001 ha 3 spazi di indirizzamento,

Address Hex	Bus P	Bus X	Bus Y
0000-0100	Internal RAM	Internal RAM	Internal RAM
0100-0200	Internal RAM	–	External RAM
0200-0300	External RAM	CCDSEQ bus	External RAM
0300-4000	External RAM	–	External RAM

Tabella 3: Mappa di indirizzamento del DSP 56001. Il DSP ha 3 spazi di indirizzamento, uno per il programma, e due per i dati (X e Y).

Address Hex	Card	Read	Write
0	CCDSEQ	–	Status A
1	CCDSEQ	–	Status B
2	CCDSEQ	ID + Sync	–
3	CCDSEQ	–	Led
10-13	CKBUF	–	Level DAC ch. 1-4
14	CKBUF	–	Bias DAC
15	CKBUF	–	Reset DAC
18	CKBUF	telemetry ADC data	ADC control
1C	CKBUF	–	Fast pixel shift
1D	CKBUF	–	Fast line shift
80	ADC_GAL 1	ADC ch1 value	ADC ch1 link control
81	ADC_GAL 1	ADC ch2 value	ADC ch2 link control
82	ADC_GAL 2	ADC ch3 value	ADC ch3 link control
83	ADC_GAL 2	ADC ch4 value	ADC ch4 link control

Tabella 4: Mappa di indirizzamento del CCDBUS.

uno dedicato al programma (bus P) e due per i dati (X e Y). Il bus del CCDSEQ è mappato sul bus dati X, a partire dalla locazione 0x200.

Il bus CCDSEQ permette di accedere a 256 locazioni di I/O. Sono utilizzati solo i 16 bit meno significativi del bus dati, in quanto gli 8 bit più significativi sono utilizzati dalla logica di delay. Pertanto in scrittura occorre accertarsi che questi 8 bit siano posti a zero, se non si intende usare wait states. Il DSP legge comunque parole a 24 bit, e pertanto gli 8 bit più significativi di una lettura sono sempre indefiniti, e andrebbero mascherati a zero dopo ogni lettura.

In tabella 4 sono elencate le locazioni di memorie utilizzate nell'hardware di NICS.

Il sistema è composto da 4 schede visibili sul bus: il controller, 2 schede di ADC sincrone (GAL.ADC1 e 2) e una scheda di clock buffer.

Le schede di ADC contengono ciascuna due convertitori. Il valore convertito può essere letto dal DSP (anche se questa possibilità non viene utilizzata), e viene comunque serializzato su di un link TRAM.

È possibile scrivere una quantità arbitraria a 16 bit su ciascun link indirizzando la locazione corrispondente sul bus del CCDSEQ. In questo modo possono essere generati protocolli complessi sul link. Il software attuale utilizza un protocollo a lunghezza variabile, in cui il numero di parole attese sul link va scritto in anticipo sul link stesso. Pertanto il DSP scrive su ogni link utilizzato il numero di conversioni previste prima della scansione di ogni riga. La routine `M_SCAN` è stata modificata in modo da scrivere questo numero prima di generare la sequenza (waveform) corrispondente alla lettura di una riga. Per permettere alla logica di DMA del transputer di processare questo numero, la routine aspetta un tempo prefissato prima di iniziare a generare la



sequenza.

La scheda di clock buffer permette di controllare 6 convertitori DAC e un convertitore ADC posti sulla scheda di condizionamento analogico del segnale. I convertitori sono controllati tramite un'interfaccia seriale realizzata in un chip Xilinx. La sequenza di scrittura sul questa interfaccia seriale viene iniziata da una scrittura alla locazione corrispondente sul bus. La scala dei convertitori è di 1 mV per unità (max. 4.095 V) per i convertitori di sottrazione della continua e per la tensione di reset, mentre il bias del substrato ha una scala di 0.3 mV/unità (max. 1.23 V).

Impostando un comando di start conversion al convertitore ADC, si può leggere il valore di tensione generato da uno degli ADC, o della tensione di alimentazione della scheda. Il valore viene immagazzinato in un registro della scheda, e può essere letto successivamente dal DSP.

## 4 Struttura del programma

Salvo dove espressamente indicato, viene descritto il programma nella versione per NICS, incluse quindi tutte le modifiche fatte rispetto al software originale CCDWG.

Il programma consiste in un singolo loop di esecuzione, che accetta un comando, esegue l'operazione richiesta, e ritorna un pacchetto di telemetria.

Prima di interpretare il comando, la routine copia il buffer di comandi nel buffer di telemetria. Tutte le funzioni sono svolte da una subroutine, che provvede eventualmente a modificare il contenuto del buffer di telemetria. Al termine del loop, il buffer di telemetria viene rispedito al transputer.

La maggior parte delle funzioni richiede una subroutine relativamente breve per l'esecuzione del comando. La subroutine si chiama con il nome simbolico assegnato al comando.

La routine che esegue il comando di load waveform è stata modificata, in quanto un bug nel software originale impediva di caricare più di 6 waveform consecutivamente.

La routine di test waveform utilizza una routine di servizio, che invia una sequenza di waveform sul CCDSEQ bus. Questa routine viene impiegata anche dalla routine che esegue il comando di integrazione.

Il comando di integrazione esegue una ripetizione per il numero di cicli richiesto di una sequenza di reset e lettura del CCD, attesa per il tempo specificato, e lettura non distruttiva del CCD. Per ora non sono previsti modi più intelligenti di lettura. La routine chiama una subroutine di lettura generica dell'area specificata, specificando la waveform da usare per la lettura dei pixel. Pertanto risulta semplice creare modi differenti di funzionamento, in futuro.

## 5 Descrizione dettagliata delle routines

In questo capitolo sono descritte in dettaglio le procedure utilizzate dai vari comandi.

### 5.1 Get telemetry

Il comando esegue una serie di letture sull'ADC posto nella scheda di condizionamento dei segnali analogici. Vengono letti in successione i valori delle tensioni di plateau dei 4 canali del multiplexer, della tensione di bias, della tensione del segnale di reset, e delle due alimentazioni della scheda. Al termine della misura, viene ritornato un pacchetto di telemetria con 8 valori.

## 5.2 Dump del bus X

Il comando utilizza la facility di modifica dei wait states del DSP. Il registro di wait viene modificato in modo da aggiungere 2 wait states prima di ogni lettura sul bus X. Il registro viene resettato al valore di default (0 wait states) al termine della routine.

## 5.3 Load waveform table

Il comando confronta l'ID della waveform specificata nei parametri con i valori previsti, elencati nel cap. 2.3.4. La tabella viene memorizzata in un buffer di memoria allocato da un'area riservata allo scopo. Non viene deallocato un eventuale buffer usato da una waveform caricata in precedenza con lo stesso ID, in quanto questo comporterebbe una complicazione eccessiva del programma.

Il software originale conteneva un bug: la ricerca dell'ID viene fatta con una istruzione di loop hardware del DSP. Quando l'ID corretto veniva trovato, non veniva eseguita, come richiesto, un'istruzione di pulizia dello stack, che quindi si esauriva dopo 6 esecuzioni del comando.

## 5.4 Test waveform

Utilizza direttamente la subroutine `M_SCAN` descritta nel cap. 5.6, che esegue le waveform durante l'integrazione.

## 5.5 Start integration

È il comando che realizza l'integrazione vera e propria.

Nella versione attuale, il terzo parametro (modo di lettura del sensore) non viene utilizzato, e l'unico modo implementato è la lettura a doppio campionamento. Il sensore viene letto due volte, la prima contestualmente ad una operazione di reset, e la seconda al termine dell'integrazione.

Dopo aver salvato i parametri, la routine esegue una prima lettura dell'area specificata del sensore, previo reset di ciascun pixel. I pixel non specificati (al di fuori dell'area specificata con il comando `SetBox`) *non* vengono resettati.

Quindi viene eseguito un loop di ritardo. Il loop viene eseguito in software, ripetendo una moltiplicazione per un numero di volte proporzionale al tempo di integrazione specificato. Il numero di cicli è ottenuto moltiplicando il numero specificato di millisecondi per una costante, determinata empiricamente. La moltiplicazione viene eseguita a 56 bit, e quindi non vi è rischio di overflow se si specificano tempi di integrazione molto elevati.

Infine il sensore è letto una seconda volta, in modo non distruttivo. L'intera procedura viene quindi ripetuta il numero di volte specificato.

Sono possibili diversi miglioramenti, che richiedono però un software più complesso anche dal lato dei transputer. Il più diretto consiste nell'utilizzare una sequenza di lettura/cancellazione/lettura di ciascun pixel in tutte le sottointegrazioni eccetto l'ultima, per ridurre i tempi morti.

La routine utilizza una subroutine di lettura del sensore che accetta come parametro la waveform da usare effettivamente per la lettura dei pixel di interesse. In questo modo è possibile realizzare sequenze di lettura differenti, con campionamento multiplo, ecc., in modo relativamente semplice.

### 5.5.1 Routine di lettura del sensore

Questa routine accetta come parametro la waveform da utilizzare per l'inizializzazione della riga, e quindi può essere utilizzata per il reset, la lettura o la doppia lettura dell'area di interesse.

La routine esegue una serie di sequenze di waveform di inizio frame, avanzamento riga, inizio riga, avanzamento pixel, in modo da posizionarsi all'inizio della box specificata dai comandi Set Area e Set Box. La waveform da utilizzare per l'inizio riga è specificata come parametro, in modo da poter usare la stessa routine per letture sia non distruttive che distruttive. Quindi la routine esegue una serie di waveform di lettura pixel, e avanzamento riga, in modo da leggere l'area specificata.

La routine utilizza estensivamente la subroutine di generazione di una waveform

## 5.6 Routine di generazione di una waveform

Questa routine richiede in ingresso l'indirizzo di una waveform, e il numero di volte che la waveform elementare viene ripetuta. Calcola il numero di conversioni generate dalla waveform, utilizzando l'assunzione che le waveform di clear e di lettura generino una conversione per pixel, mentre quella di lettura/clear/lettura ne generi due. Le rimanenti waveform non generano conversioni.

La routine inizializza i link degli ADC scrivendo su ciascuno di essi il numero di word (conversioni) attese, e quindi copia la tabella di waveform sulla porta specificata nella waveform stessa, per il numero di volte indicato nel parametro.

## Riferimenti bibliografici

- [1] Bortoletto F., et al. (1992): "Architecture of the Galileo CCD controllers", in Fusi Pecci F., Stirpe G.: "TNG Instrument Plan: A progress report", pagg. 61–84, Osservatorio Astronomico di Bologna, (1992),
- [2] Bortoletto F. et al. (1992): "CCD Sequencer User Manual", rapp. int. Osservatorio di Padova
- [3] Lisi et al. (1994): "The IR Imager/Spectrograph", in Fusi Pecci F., Stirpe G. Zitelli V.: "TNG Instrument Plan: II. A progress report", Osservatorio Astronomico di Bologna, WWW: <http://www.bo.astro.it/galileo/galbo.html>
- [4] Comoretto G., Baffa C., Lisi F.: "NICS–Camera infrarossa TNG - Descrizione generale dell'elettronica di controllo e presa dati", Osservatorio di Arcetri, rapporto interno n. 4/95 (1995).

# Indice

<b>1</b>	<b>Specifiche di progetto</b>	<b>1</b>
<b>2</b>	<b>Comunicazione con il DSP</b>	<b>2</b>
2.1	Protocollo di comunicazione standard CCDWG . . . . .	2
2.2	Comandi definiti nel protocollo CCDWG . . . . .	3
2.2.1	Ask telemetry . . . . .	3
2.2.2	Set location . . . . .	3
2.2.3	Test waveform . . . . .	3
2.2.4	Dump X (Y) bus . . . . .	4
2.2.5	Return card ID and Sync bit . . . . .	4
2.2.6	Set sensor dimensions, and reaout box . . . . .	4
2.2.7	Load waveform table . . . . .	4
2.2.8	Start integration . . . . .	5
2.3	Aggiunte e modifiche nella versione per NICS . . . . .	5
2.3.1	Get Telemetry . . . . .	6
2.3.2	Dump from X bus . . . . .	6
2.3.3	Set sensor dimensions . . . . .	6
2.3.4	Load waveform table . . . . .	6
2.3.5	Start integration . . . . .	6
<b>3</b>	<b>Spazio di indirizzamento del DSP</b>	<b>6</b>
<b>4</b>	<b>Struttura del programma</b>	<b>8</b>
<b>5</b>	<b>Descrizione dettagliata delle routines</b>	<b>8</b>
5.1	Get telemetry . . . . .	8
5.2	Dump del bus X . . . . .	9
5.3	Load waveform table . . . . .	9
5.4	Test waveform . . . . .	9
5.5	Start integration . . . . .	9
5.5.1	Routine di lettura del sensore . . . . .	9
5.6	Routine di generazione di una waveform . . . . .	10